



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### **DIT203 Logic in Computer Science, 7.5 credits**

Matematisk logik för datavetenskap, 7,5 högskolepoäng

*Second Cycle*

---

#### **Confirmation**

This course syllabus was confirmed by Department of Computer Science and Engineering on 2022-11-17 to be valid from 2023-08-28, autumn semester of 2023.

*Field of education:* Science 100%

*Department:* Department of Computer Science and Engineering

#### **Position in the educational system**

The course is a part of the Computer Science Master's programme and an elective course at the University of Gothenburg.

The course can be part of the following programmes: 1) Mathematical Sciences, Master's Programme (N2MAT), 2) Computer Science, Master's Programme (N2COS), 3) Computer Science, Bachelor's Programme (N1COS) and 4) Bachelor's Programme in Mathematics (N1MAT)

#### *Main field of studies*

Mathematics

Computer Science

#### *Specialization*

A1N, Second cycle, has only first-cycle course/s as entry requirements

A1N, Second cycle, has only first-cycle course/s as entry requirements

#### **Entry requirements**

To be eligible for the course, students must have successfully completed courses corresponding to 105 hec within the subject Computer Science or Mathematics, including 7.5 hec in discrete mathematics (for example DIT980 Discrete Mathematics for Computer Scientists or the sub-course Introductory Algebra of MMG200 Mathematics 1).

Applicants must prove knowledge of English: English 6/English B or the equivalent level of an internationally recognized test, for example TOEFL, IELTS.

### **Learning outcomes**

After completing the course the student is expected to be able to:

#### *Knowledge and understanding*

- explain when a given formula is a tautology
- explain the notion of model of a first-order language and of temporal logic
- explain when a first-order and a temporal logic formula is semantically valid
- explain how to check if a branching-time temporal logic formula is valid in a given model
- explain the meaning of the soundness and completeness theorems for propositional and predicate calculus

#### *Competence and skills*

- write and check proofs in natural deduction for propositional and predicate calculus
- specify properties of a reactive system using linear-time temporal logic and branching-time temporal logic

#### *Judgement and approach*

- judge the relevance of logical reasoning in computer science, i.e. for modelling computer systems
- analyse the applicability of logical tools to solve problems in computer science, i.e. finding bugs with the use of model checking

### **Course content**

Powerful tools for verifying software and hardware systems have been developed. These tools rely in a crucial way on logical techniques. This course provides a sound basis in logic and a short introduction to some logical frameworks used in modelling, specifying and verifying computer systems. A sound basic knowledge in logic is a welcome prerequisite for courses in program verification, formal methods and artificial intelligence.

The course covers propositional and predicate calculus, and model-checking. More concretely, the course gives a thorough introduction to fundamental notions of logic such as natural deduction, semantics of both propositional and predicate calculus, soundness and completeness, conjunctive normal forms, Horn clauses, undecidability and expressiveness of predicate logic, plus an introduction to model checking: Linear-

time temporal logic (LTL) and Branching-time temporal logic (CTL).

*Sub-courses*

**1. Written hall examination** (*Skriftlig salstentamen*), 7.5 credits

Grading scale: Pass with distinction (5), Pass with credit (4), Pass (3) and Fail (U)

**Form of teaching**

The course consists of lectures, exercise sessions and non-obligatory assignments.

*Language of instruction:* English

**Assessment**

The course is examined by an individual written exam taking place in an examination hall at the end of the course.

There will be non-obligatory individual assignments offered during the course which grant bonus points for the written exam. These bonus points are valid during the whole academic year.

If a student, who has failed the same examined component twice, wishes to change examiner before the next examination, a written application shall be sent to the department responsible for the course and shall be granted unless there are special reasons to the contrary (Chapter 6, Section 22 of Higher Education Ordinance).

In cases where a course has been discontinued or has undergone major changes, the student shall normally be guaranteed at least three examination occasions (including the ordinary examination) during a period of at least one year from the last time the course was given.

**Grades**

The grading scale comprises: Pass with distinction (5), Pass with credit (4), Pass (3) and Fail (U).

The final grade of the course is based on the result of the written exam.

**Course evaluation**

The course is evaluated through meeting during and after the course between teachers and student representatives. Further, an anonymous questionnaire is used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.

**Additional information**

The course is a joint course together with Chalmers.

Course literature will be announced the latest 8 weeks prior to the start of the course.

The course replaces the course DIT202 Logic in Computer Science, 7.5 credits. The course cannot be included in a degree which contains DIT202. Neither can the course be included in a degree which is based on another degree in which the course DIT202 is included